Port scan is not for pussies

Know yourself, know your enemy

Fred RAYNAL
fraynal@quarkslab.com
@fredraynal

Adrien GUINET
aguinet@quarkslab.com
@adriengnt

QUARKSLAB
INNOVATIVE SECURITY

How did it start?

I want to scan the Internet!!!

- Scan for obscure web forums to gather versions of phpBB, vBulletin and others
- Scan for card sharing servers
- Get carrot juice, a veggie burger and some sleep
- Idea: scan for everything everywhere
- Internet Census (2012): well played, f*****g Carna Botnet

Why do we care about network recon?

Motivation

- For attackers: information is as valuable as 0 days
  - Allow to build the attack path
  - Avoid wasting 0 days
  - Find opportunistic targets
- For defenders: learn about yourself
  - *Should* allow to learn about their own attack surface
  - *Should* guide to concentrate defenses where one is the most exposed and sensitive

Roadmap

This talk

- Engineering: how to design an Internet wide scanner
- Targeting: what is a target?
- Applications: what we find on the Internet

Plan

Plan

Tools of the trade

Well known tools for pentesters

- Port scanners: nmap, zmap, masscan...
- Banner grabbers++: snmpwalk, sslscanner, nikto, BlindElephant, ...
- OS fingerprinting: nmap, pOf, sinFP...

Tools of the trade

Well known tools for pentesters

- Port scanners: nmap, zmap, masscan...
- Banner grabbers++: snmpwalk, sslscanner, nikto, BlindElephant, ...
- OS fingerprinting: nmap, pOf, sinFP...

Problems

- Distribution and scalability
- No searchable web interface

Hmm, it looks like Vulnerability scanner?

It looks like, but it does not taste like!

- Try to scan a /B with Qualys / Nessus / *
  - Expensive: need to sell your kid's kidneys at least
  - Super slow: imagine the 1,000,000+ page PDF report
- Might do something with Metasploit
  - Add an efficient port scanner
  - Add a database and index the results

Hmm, it looks like Vulnerability scanner?

It looks like, but it does not taste like!

- Try to scan a /B with Qualys / Nessus / *
  - Expensive: need to sell your kid's kidneys at least
  - Super slow: imagine the 1,000,000+ page PDF report
- Might do something with Metasploit
  - Add an efficient port scanner
  - Add a database and index the results

Problems: size matters

- Costs
- SCALING again!

Plan

Needs and objectives

What we want

- Collect L7 unstructured information: texts, certificates, images, keys...
- Analyze all the unstructured information

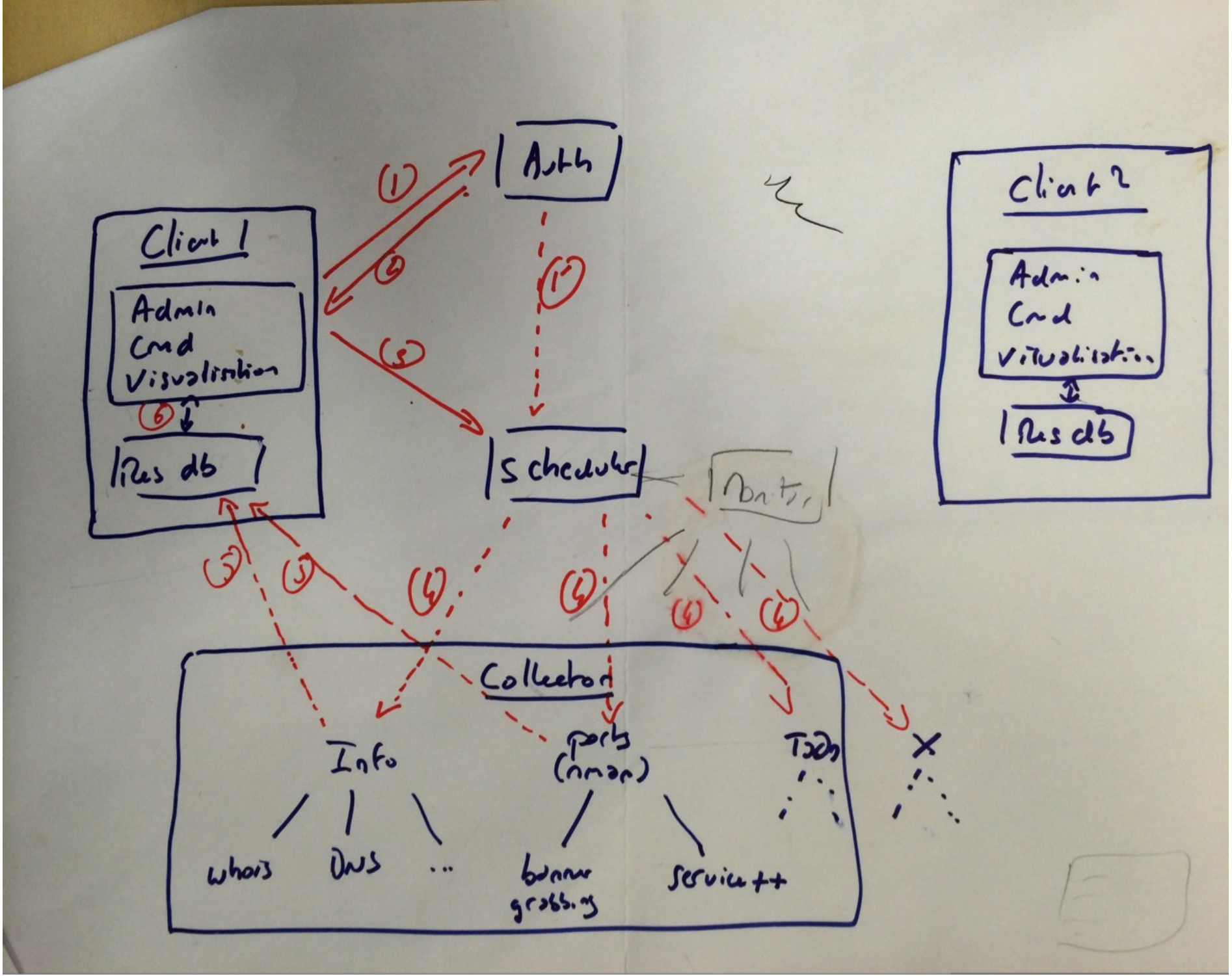
How to get it

- Distribute multiple scans among multiple probes
- Thin probes: "local" view of the scan, they only know what they scan, nothing else
- Dynamic scalability:
  - Add/remove targets on the fly
  - Add/remove probes on the fly

Design: piece of cake!

KISS = Keep It Simple, Stupid

- Use a port scanner and a few other tools
- Distribute the scan job among *n* machines with for instance RabbitMQ
- Gather the data in a big database


Auth
Client 1
Admin
Cmd
Visualisation
Res db
Scheduler
Monitor
Client 2
Admin
Cmd
visualisation
Res db
Collector
Info
ports
(nmap)
T2Ds
X
whois
DNS
...
banner
grabbing
service++

I LOVE IT WHEN A PLAN COMES TOGETHER.

Plan

At first, we had nmap

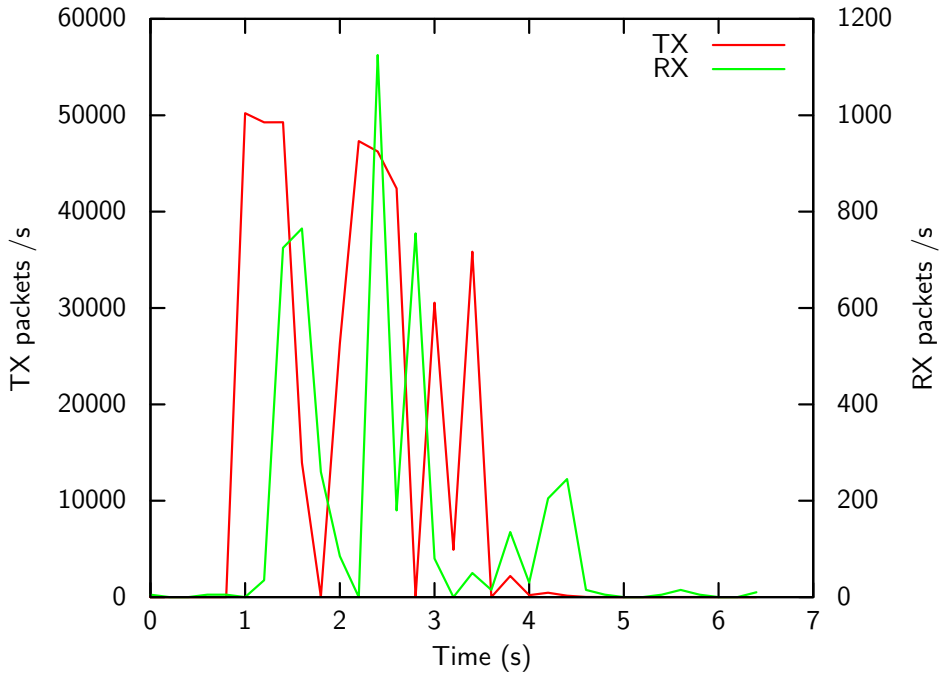
Pros

- Stable and widely used
- Powerful NSE scripts engine
- Correctly fast with good timing options

Cons

- Runs on a single host
- Can not add target on the fly (even with -iL -)

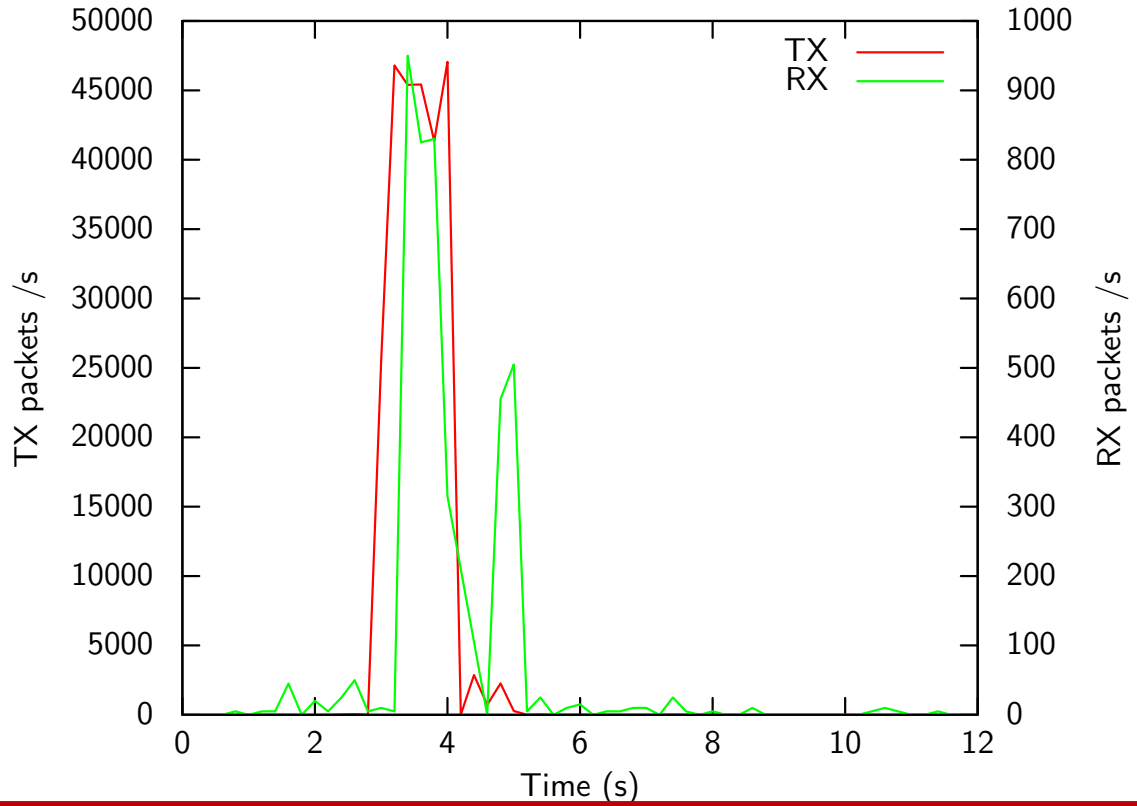
At first, we had nmap



Remarks

- Scan targets by group and wait for answers
- Multiple "waiting" sessions
- This is where masscan and zmap are somehow better

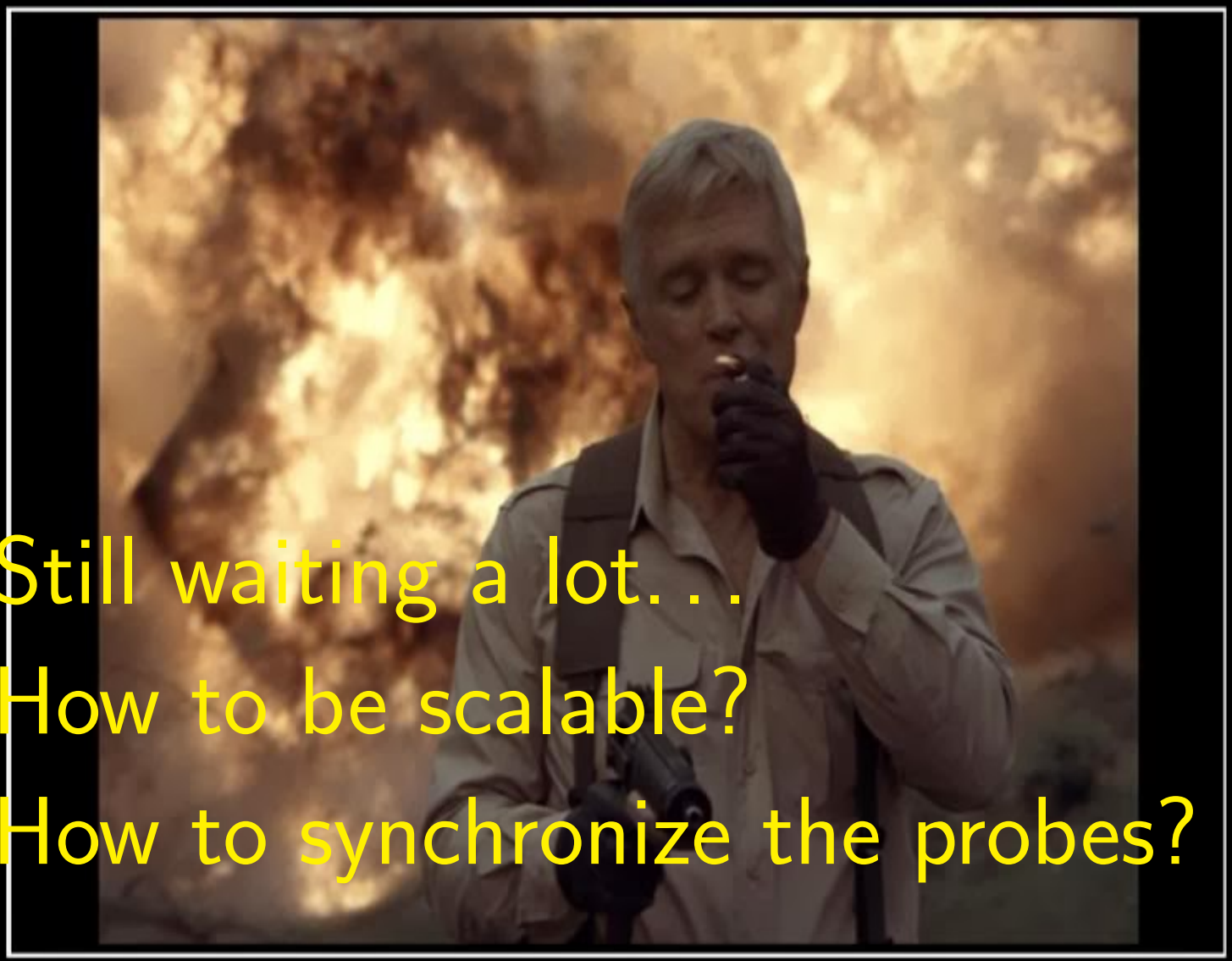
Multiple nmap: one to rule them all?



On a single host

- Network exhaustion
- Process limitation
- No synchronization between the processes

⇒ Worst on multiple hosts!



- Still waiting a lot...
- How to be scalable?
- How to synchronize the probes?

I LOVE IT WHEN A PLAN COMES TOGETHER.

Becoming scalable: a first try

Examples with 3 probes

- Divide the target set in 3
- Give each host a third of the target space
- Collect the results from the probes

- Can not add new targets
- Can not add new probes

$\Rightarrow$ This is not scalability

I LOVE IT WHEN A PLAN COMES TOGETHER.

Becoming scalable: the plan B

Being scalable

- Divide the target set in fixed-size randomized blocks of IPs/blocks
- Create a queue of tasks to perform
- Send them to your probes on-demand

Scalability 101: what we need

- A message passing protocol (rabbitmq, mpi, ...) to give orders and get back the results
- A scanner (nmap for now)
- Something to keep track of what's been done

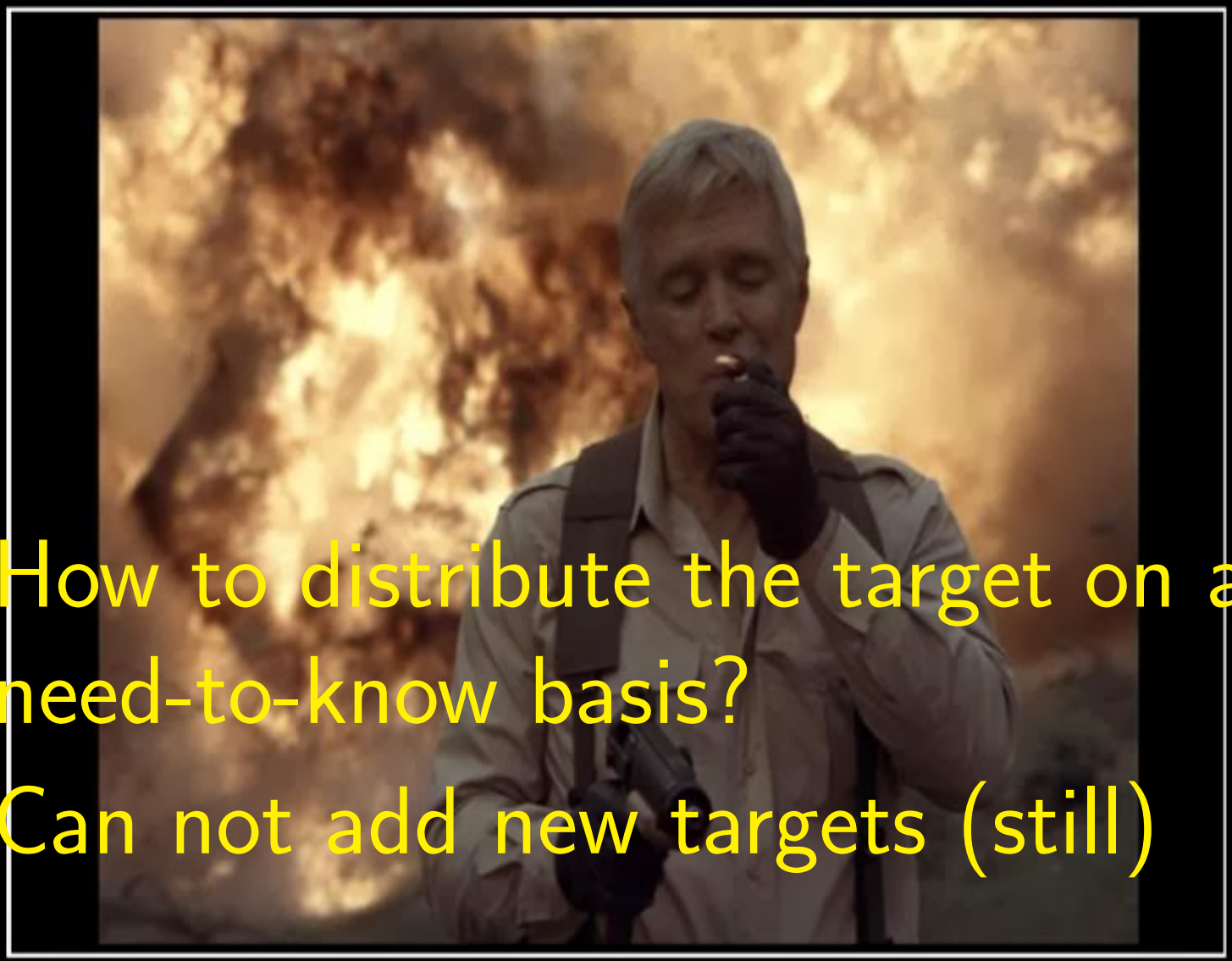
Becoming scalable: the plan B (what we need)

Another piece of cake

- A library that randomize the target set
- AMQP for the task management and tracking

Extra-bonus

- Probes are on a need-to-know basics
- New probes can be added on the fly, they just grab new tasks
- Probes can get away without ACKing a task, it will be performed by a new one



- How to distribute the target on a need-to-know basis?
- Can not add new targets (still)

I LOVE IT WHEN A PLAN COMES TOGETHER.

Splitting the targets

What is a target?

- A target is a union / exclusion of intervals of IP addresses

Naive algorithm

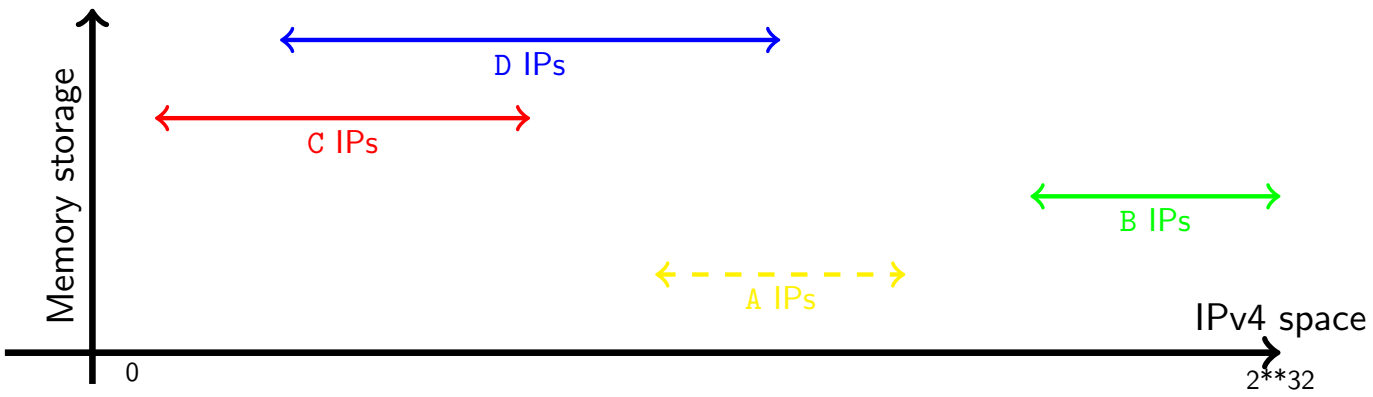
- Create a list of all unique IP addresses
- Randomize the set to avoid consecutive scanning (thus complains)


2^{32} IP * 4 bytes = 16Gb in RAM :(
I LOVE IT WHEN A PLAN COMES TOGETHER.

Splitting the targets with a PRNG

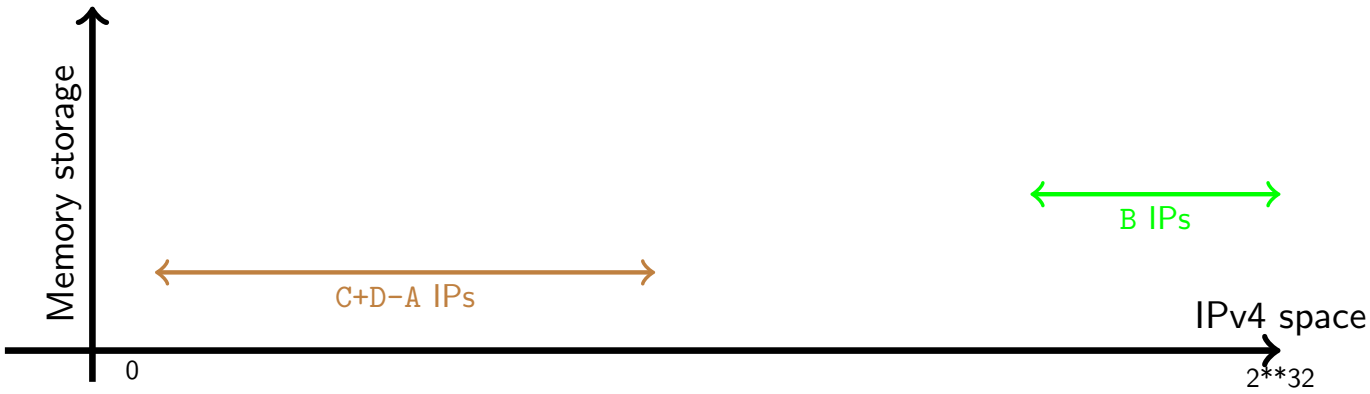
Step 1: initial configuration

- Wanted ranges are the full lines
- Excluded ranges are the dashed lines



Splitting the targets with a PRNG

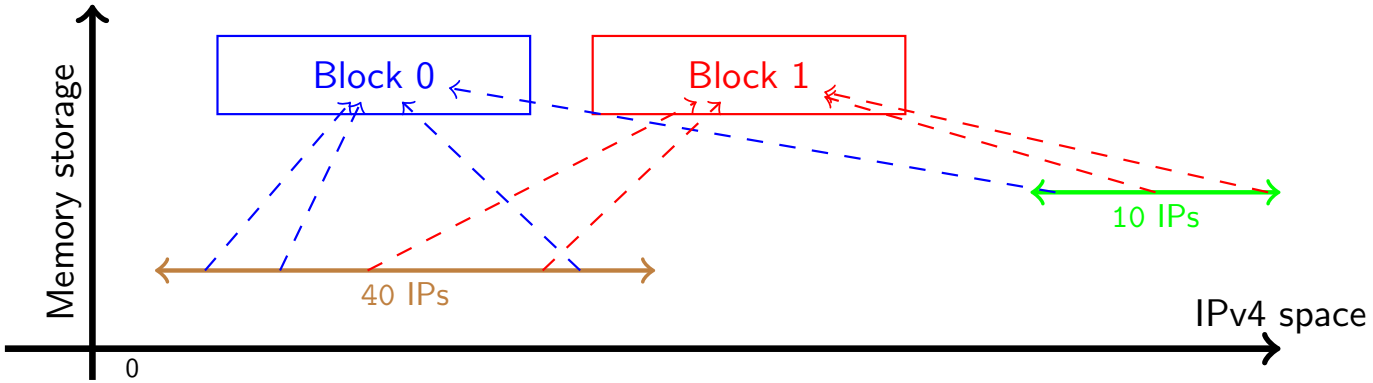
Step 2: sorting and merging intervals



Splitting the targets with a PRNG

Randomization

- There are `N (=C+D+B-A)` IPs among `R (=2)` distinct ranges
- Compute a random permutation of [0..N[
- For each integer `i` of this permutation, grab the IP at the i-th index
- Create blocks of `G` (=4 for instance) randomly choosen IPs and send them to the probes
- An example: [30, 10, 5, 42, 20, 28, 48, 49, ...]



This is where we are now thanks to distribution of the scan

$\Rightarrow$ Time for optimization!

Plan

Optimization: upgrade the scanner

zmap

- Asynchronous I/O engine for the packets
- Can share a target on several hosts
- Can not add probes dynamically
- Can not add targets on the fly
- Scripting is a pain
- Requires a Telco for a maximum efficiency

masscan

- Asynchronous I/O engine for the packets
- Can share a target on several hosts
- Can not add probes dynamically
- Can not add targets on the fly
- Scripting is a pain++
- Requires a Telco for a maximum efficiency

Keep in mind...

Scanning very large sets of IPs dynamically is not only about sending packets as fast as possible...

The Devil is in the details!

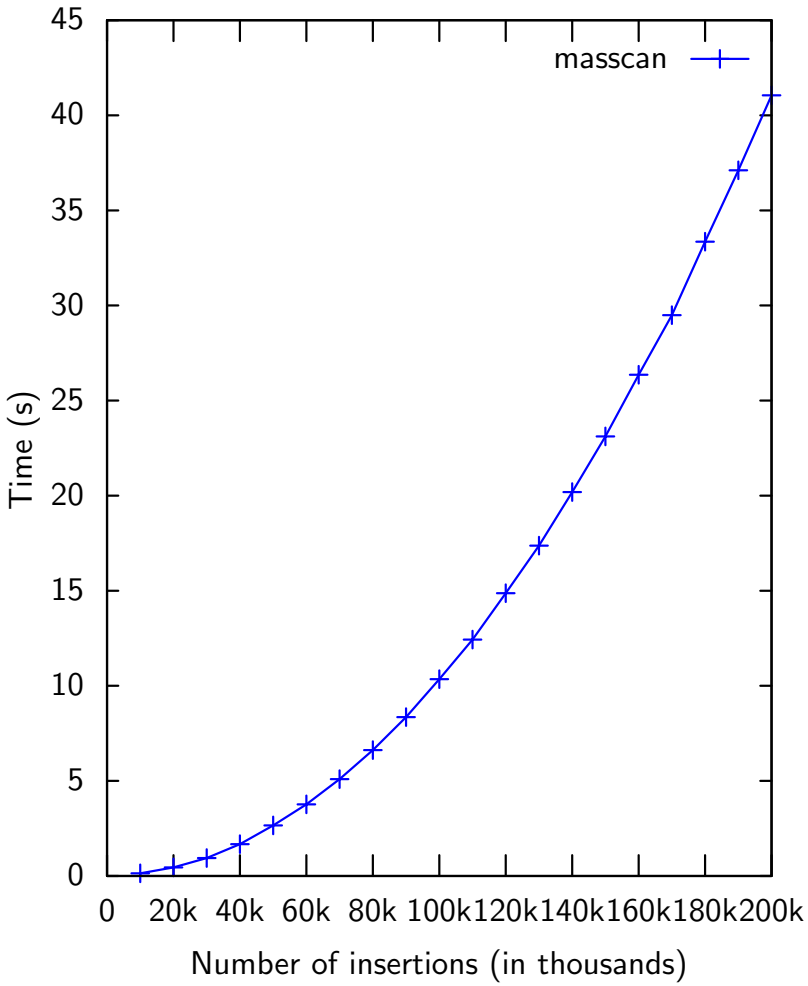
Scanning the results of a scan

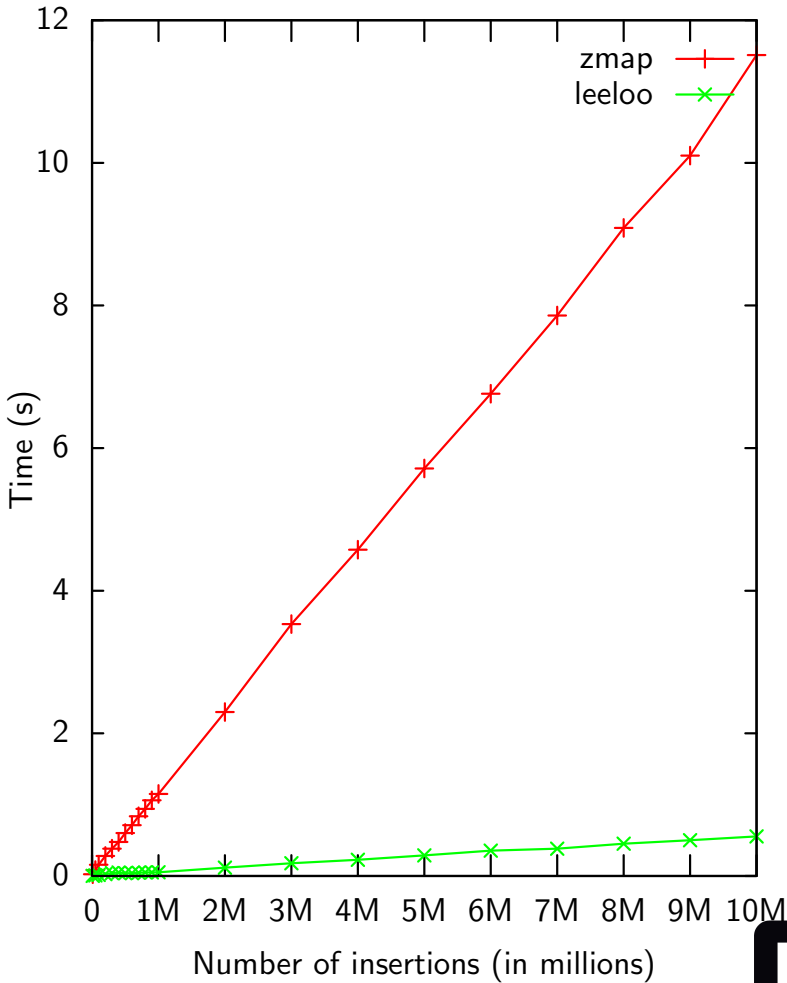
- You scan a large set of IPs
- You sort the result according to whatever criteria (port 1234 open)
- You want to rescan this subset
- Problem: you now have like 200k small intervals of IPs

$\Rightarrow$ Adding and looking up are complexity killing operations too

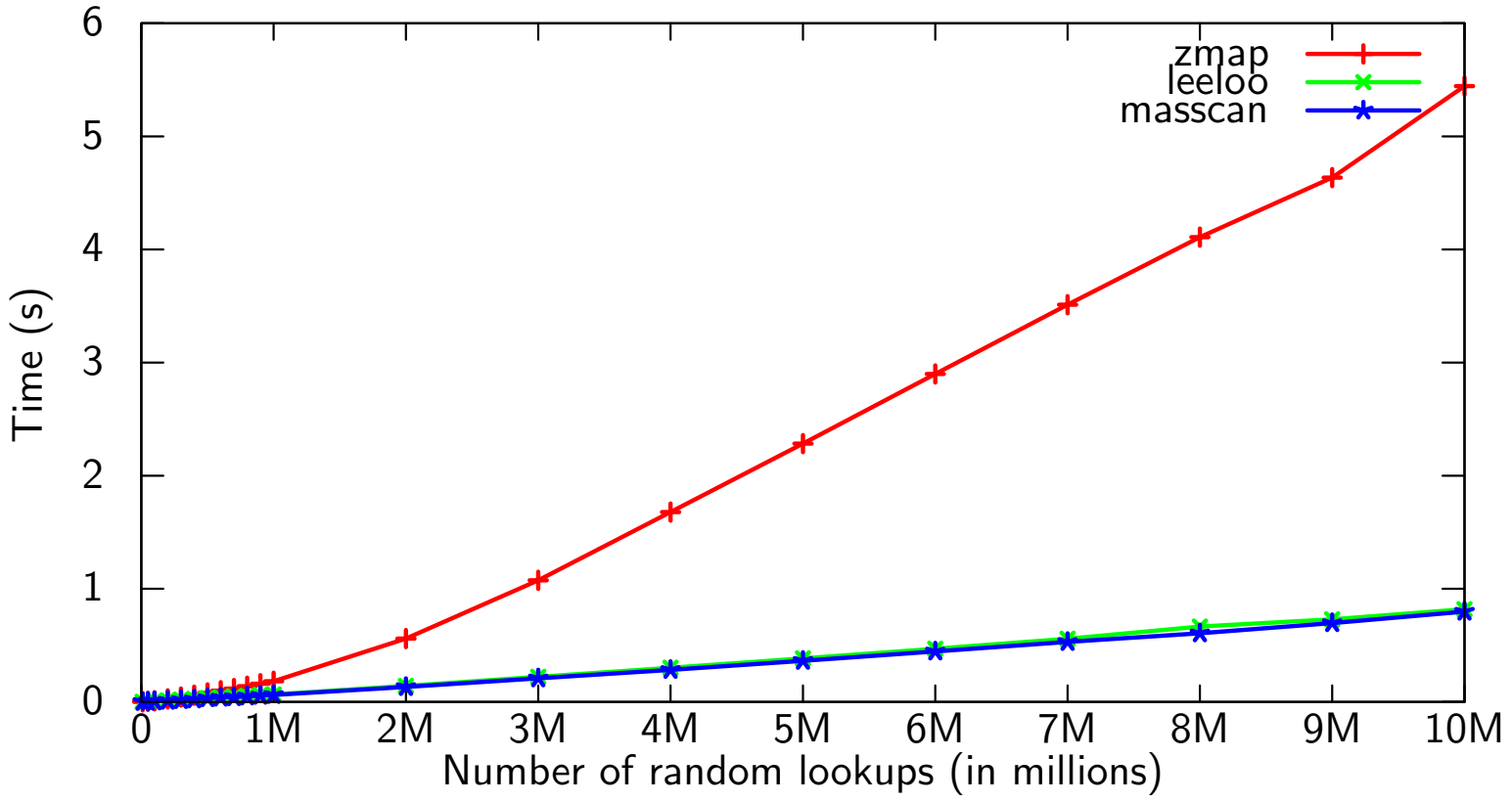
IP intervals management: intervals add performances

Benchmarks done on a Core i7-3520M





IP intervals management: random lookup performances



Lookup performances

- **liblleeloo** and **masscan** can provide about **12,204,000** random lookups/second

IP intervals management

zmap

- **Model**: intervals stored as a tree (lower memory usage), only support CIDR ranges
- **Add**: logarithmic complexity since the tree is balanced
- **Lookup**: complexity depending on the height of the tree

masscan

libleeloo

IP intervals management

zmap

masscan

- **Model**: list of intervals stored as pairs of uint32 in an array
- **Add**: exponential complexity since checking the new one is not already in a former one
- **Lookup**: logarithmic by using a pre-computed cache (non configurable size)

libleeloo

IP intervals management

zmap

masscan

libleeloo

- **Model**: same as masscan
- **Add**: just add the new intervals in the array, aggregate once at the end
- **Lookup**: logarithmic, also using a cache of configurable size (user-defined memory/performance trade-off)

The scanner of our dreams

What we dream of?

- SYN engine as efficient as masscan
- Scripting as easy as nmap
- Can run as a daemon to stream targets as they come

Patching nmap / zmap / masscan ? You have said patching?

- Need to change core components, not maintanable on a long run
- Can not support properties for IPs
- Can not support complex scan actions at layer 7

Plan

libleeloo: intervals and *properties*

libleeloo

- A **C++ library** with Python bindings
- Manage intervals of IPs as seen previously
- Support *properties*
- Available at https://github.com/quarkslab/libleeloo

Properties?

- Specific information for some IPs or ranges
- Custom TCP/UDP ports, specific credentials to test, ...

Example: using multiple properties to IPs intervals

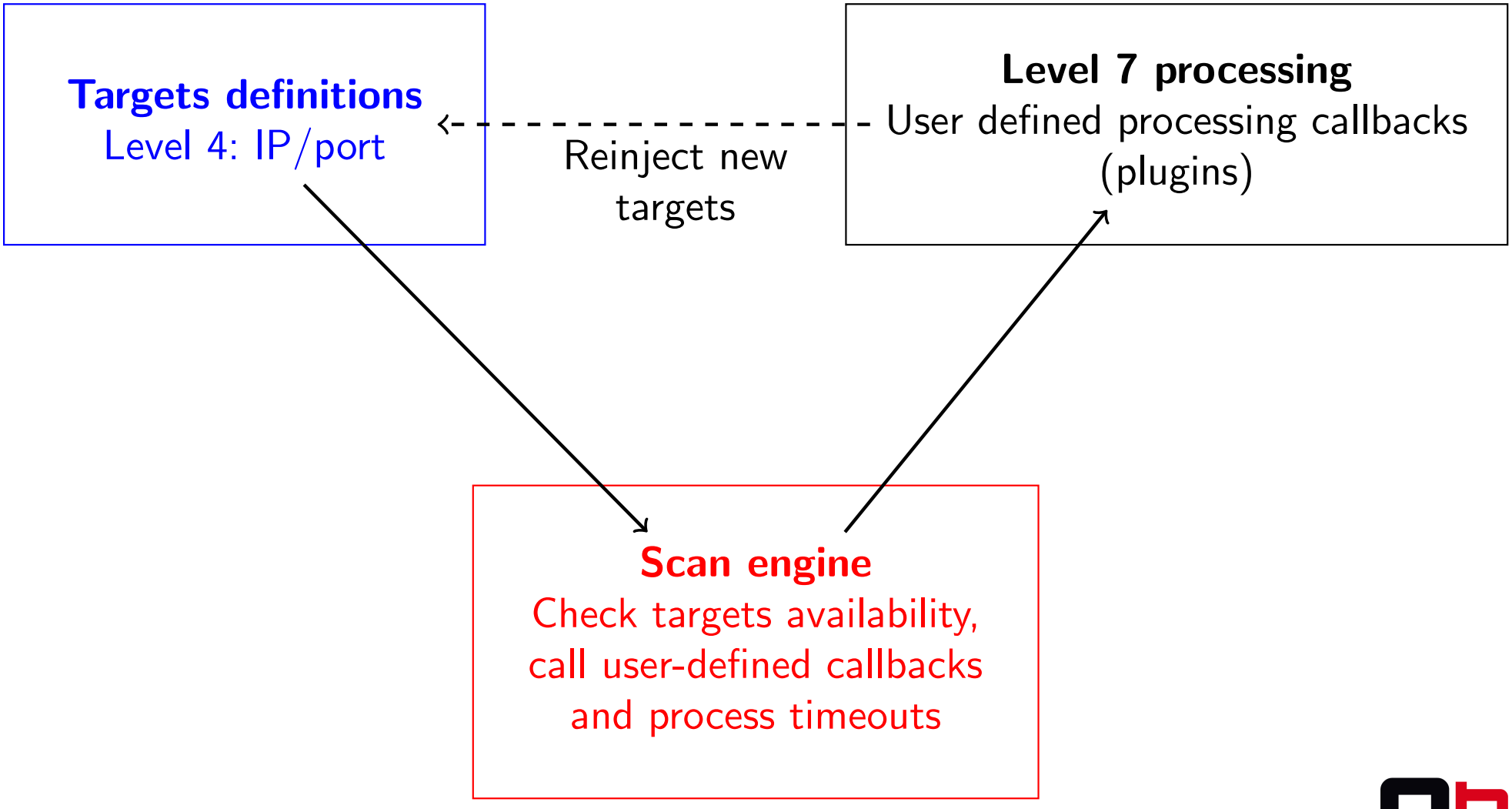
```
1  import pyleeloo
2  ranges = pyleeloo.ip_list_intervals_with_properties()
3
4  # The organisation's range
5  ranges.add("192.42.0.0/16")
6
7  # SSH servers
8  ranges.add_property("192.42.4.0/24", [22, 2222])
9  # VPN servers
10 ranges.add_property("192.42.4.10-20", [1194])
11
12 ranges.aggregate()
13
14 def merge_ports(portsA, portsB):
15   portsA.extend(portsB)
16 ranges.aggregate_properties(merge_ports)
17
18 print(ranges.property_of("192.42.66.0"))
19 >>> None
20
21 print(ranges.property_of("192.42.4.1"))
22 >>> [22, 2222]
23
24 print(ranges.property_of("192.42.4.15"))
25 >>> [22, 2222, 1194]
```

Nodescan

A L7 asynchronous engine

- A **C++ library** with Python bindings to build a custom L7 scanner
- L7 Python scripting à la nodejs with callback definitions
- Support scan pause and resume
- Allow complex actions like in SSL, SSH, SIP, ...
- Built on asynchronous UNIX sockets (for now)
- Beta on https://github.com/quarkslab/nodescan

Scanning L7 with nodescan: architecture



Scanning L7 with nodescan by example

Classical way, with a list of IPs and ports

```
 1  import pyleeloo
 2  import pynodescan
 3  from pyleeloo import tcp_port
 4
 5  ips = pyleeloo.ip_list_intervals()
 6  ips.add("37.187.47-50.70-120")
 7  ips.add("173.194.34.14")
 8
 9  ports = pyleeloo.port_list_intervals()
10  ports.add(tcp_port(80))
11  ports.add(tcp_port(22))
12  ports.add(tcp_port(443))
13
14  targets = pynodescan.IPV4TargetSet(ips, ports)
```

Scanning L7 with nodescan by example

By specifying a list of (IP, port) pairs

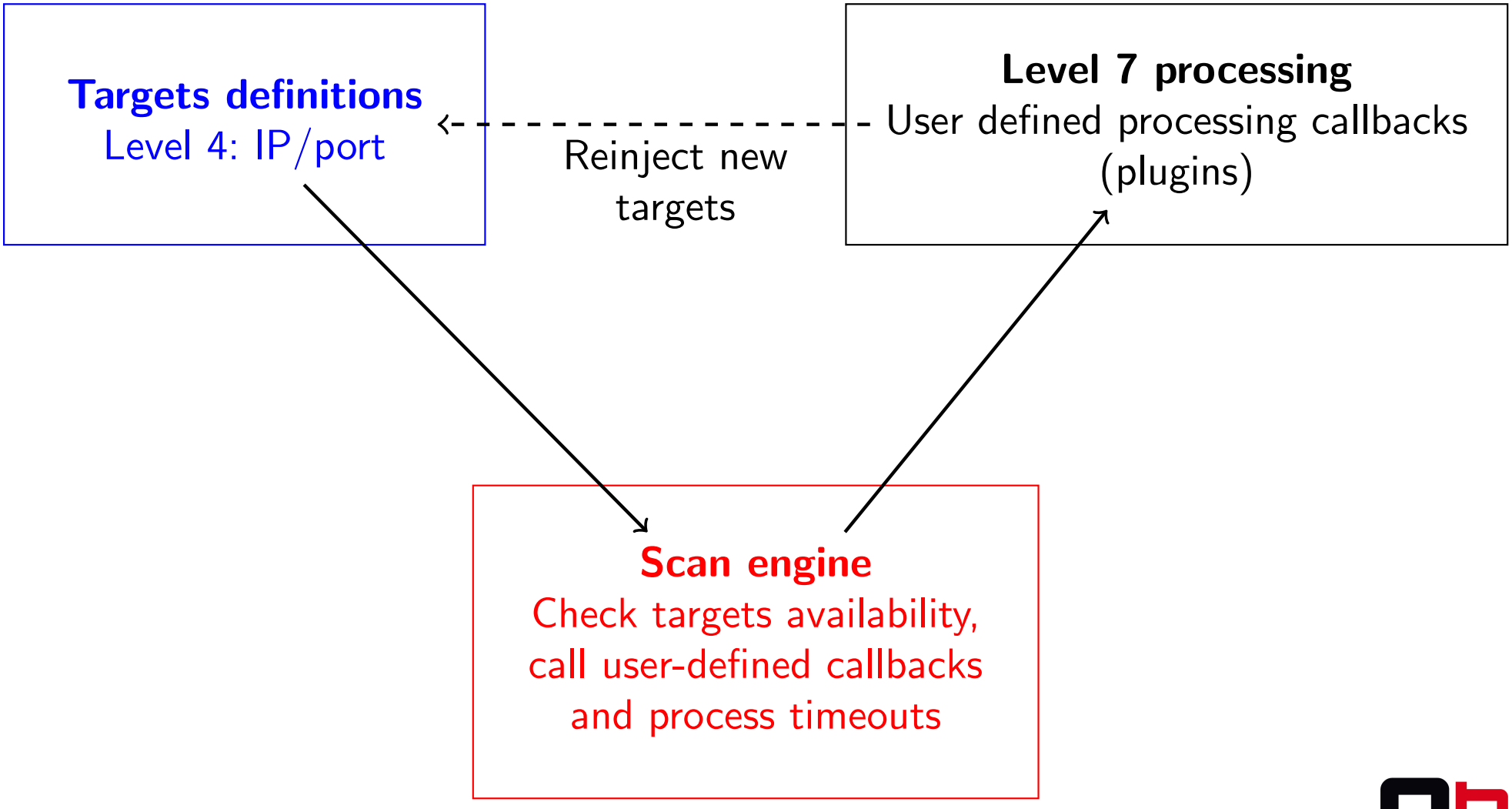
```
1  targets = pynodescan.SimpleTargetSet()
2  targets.add_target("37.187.47.70",    tcp_port(80));
3  targets.add_target("173.194.40.134", tcp_port(22));
```

Scanning L7 with nodescan by example

After the target, define how to reach them: the engine

```
1  # 'nsockets' defines the number of concurrent asynchronous
       sockets used
2  engine = pynodescan.AsyncEngine(targets=targets, nsockets
       =10000, timeout=10)
```

Scanning L7 with nodescan: architecture



Scanning L7 with nodescan by example

Simple LVL4 connection to build a HTTP scrapper

```
1  def send_payload(target, lvl4sm, hsm):
2    # Send GET /
3    target.send("GET␣/␣HTTP/1.0\n\n")
4    # Trigger on newlines
5    lvl4sm.set_char_data_trigger('\n', on_newline)
6
7    # returns True to go on with this target
8    return True
9
10 def on_newline(target, lvl4sm, hsm, buf):
11   with open("res/%d" % target.ipv4(), "ab") as f:
12     f.write(buf.tobytes())
13   return True
14
15 engine.set_lvl4_connected_callback(send_payload)
```

Scanning L7 with nodescan by example

Getting to level 7...

- Classes that wrap level 7 protocols
- Provides specific callbacks: on_content, on_certificate, ...
- User just defines what to do on each event
- Currently supports HTTP, SSH and SSL public key/certificate grabbing and SIP headers

Scanning L7 with nodescan by example

Same with HTTP wrapper

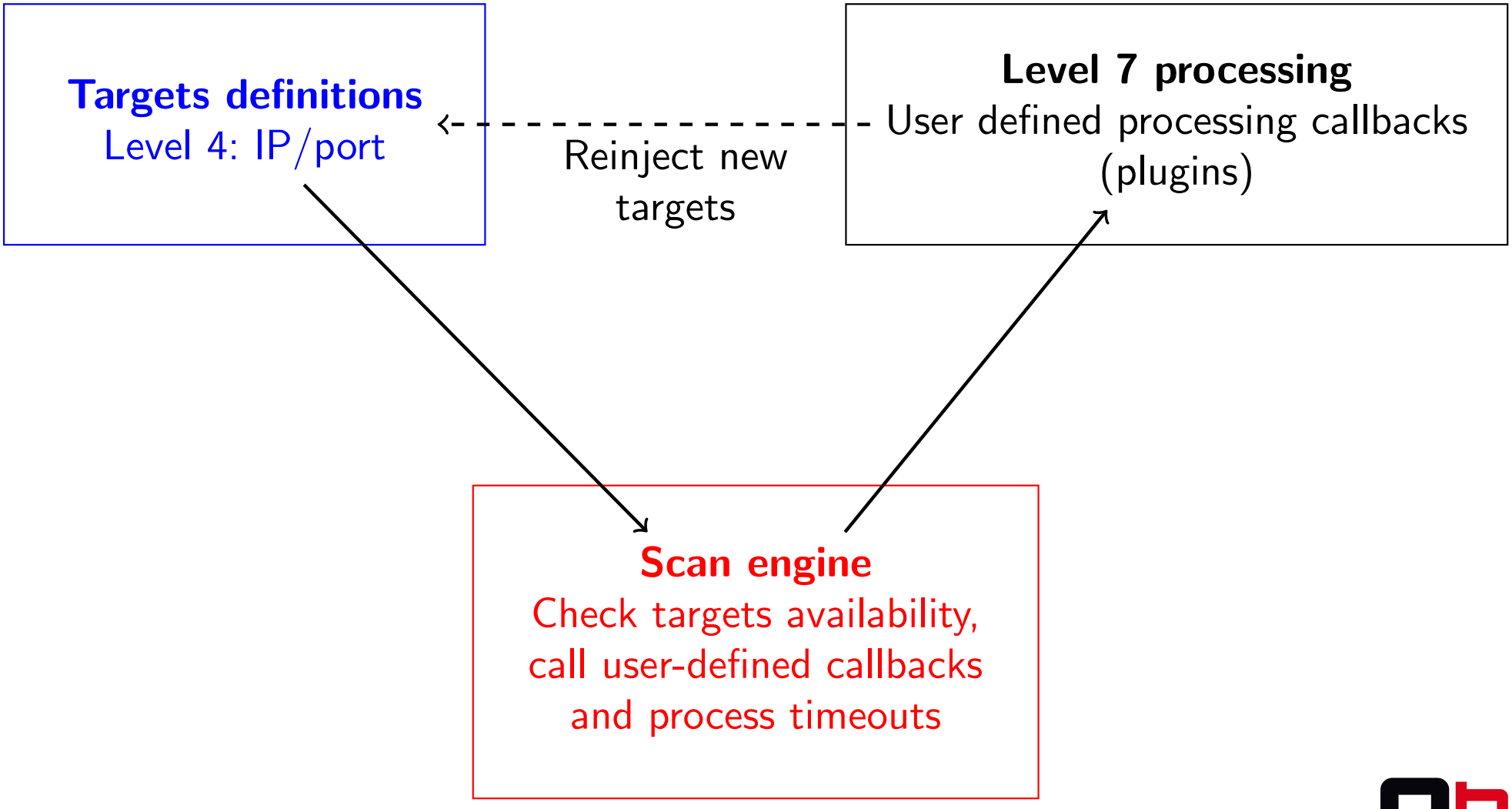
```
1  def write_header(target, key, value):
2    with open("res/%d" % target.ipv4(), "wb+") as f:
3      f.write("%s: %s\n", (key, value))
4
5  def write_content(target, code, content):
6    with open("res/%d" % target.ipv4(), "wb+") as f:
7      f.write(content.tobytes())
8
9  HTTPGrabber =
10   pynodescan.protocols.HTTPMethod("GET", "/", {"User-agent": "
         pony 1.0"})
11     .on_header(write_header)
12     .on_content(write_content)
13     .on_error(lambda target, err: print((target, err), file=sys
           .stderr))
14   )
15 engine.set_lvl4_connected_callback(HTTPGrabber)
```

Scanning L7 with nodescan by example

Try to grab SSL certificates only if the HTTP server answered

```
1   # Remember, the target set is defined as a SimpleTargetSet
2   targets = pynodescan.SimpleTargetSet()
3   targets.add("X.X.X.X/24", tcp_port(80))
4   [...]
5   HTTPGrabber = HTTPGrabber.on_content(lambda target, lvl4sm, hsm
        , content:
6       # Add a new target on the fly
7       targets.add_target(target.ipv4(), tcp_port(443)))
8   SSLGrabber = pynodescan.protocols.SSL().on_certificate(
        save_certif)
9
10  engine.set_lvl4_connected_callback(
11    PortRouter({tcp_port(80): HTTPGrabber,
12                tcp_port(443): SSLGrabber
13              }))
```

Nodescan: you have just seen the scripting



Engineering conclusion

- Scanning large sets of IPs is not only about sending raw SYN packets
- Especially if you want to do that dynamically (adding targets or probes)
- Especially if you want to collect data at layer 7 and react accordingly

Plan

Plan

Target acquisition

What is a country / company / agency in the cyberspace?

- Domains ending with the same TLD (ex.: .fr)?
- Netblocks announced at some domestic peering exchange?
- Address registry allocation?
- GeoIP?

Target = *2IP

- Convert whatever to a set of IPs
- Take GeoIP
- Take ranges from RIPE, ARIN, ...
- Take netblocks from whois databases
- Take IP behind AS

Targeting a country

Country acquisition

- Based on GeoIP
- Outsource the problem of figuring it out
- Misses some DNS names hosted overseas
- Simplify the jurisdictional issues

Country	**GeoIP**	**whois**	**GeoIP ∪ whois**
France	79M	75M	97M
Spain	29M	16M	30M

Plan

Use-case: what is *.gouv.fr

- A national sub-domain
- No specific registrar
- No general DNS

Targeting *.gouv.fr howto

Algorithm

- Find as much domains ending with *.gouv.fr as possible
- For each domain:
  - Get the corresponding IP
  - Get the whois associated to the IP
  - Consider the netrange the IP belongs to [a]

a. Assumes a hosting company might host several IPs related to *.gouv.fr

Problems / subgoals

- #1: get a whois database, which is a pain to parse
- #2: get domains from Google / Bing / other which do not want to be scrapped

Plan

whois issue: build your own reverse whois cache

Accessing whois database

- Formerly available at ipindex.homelinux.net but domain is dead now
- Bulk access to whois data has to be asked for each registrar
  - And you have to send a letter to APNIC (so 2014)

What we just need: reverse whois database

- Goal: for each IP, know to what netblock it belongs to, and who owns this netblock
- Ex.: who owns 42.0.0.0/8, 42.0.0.0/16, 42.0.0.0/24 and any potential subnetwork

Why whois servers are a pain?

whois: back to the future in the 70's

- MANY whois server, each with its output format
- Some servers answers to X.X.X.X, some to X.X.X.X/8 (and of course, not reciprocally)
- Some give inetnum of the higher level, some don't
  - whois 113.11.0.0 ⇒ inetnum: 113.11.0.0 - 113.11.127.255
  - whois 113.11.0.0/16 ⇒ inetnum: 113.0.0.0 - 113.255.255.255
  - whois 113.7.0.0/16 ⇒ inetnum: 113.0.0.0 - 113.7.255.255
  - whois 113.7.0.0 ⇒ inetnum: 113.0.0.0 - 113.7.255.255

WHAT THE FUCK

DID I JUST SEE ?

Building the reverse whois database

Algorithm

- Query every /8, /16 and /24
- Query random IP to get a granularity below /24 and aggregate the intervals

Results

- Took 1 day for all /8, /16 and /24
- Much longer for intervals below /24
- Distributed our requests, made them slowly, not to be banned
- Thank you nodescan and libleeloo :)

Plan

Getting domains: the old school way

Algorithm

- Build a list of keywords: ministere, departement, mairie, finances, ville, loi, convention, confidentiel, ...
- Query: `site:*.gouv.fr <KEYWORD>`
- Grab all domains you can

$\Rightarrow$ Got 238 domains from Bing

- Thank you http://www.tadaweb.com

Web Shopping Vidéos Images Actualités Plus Outils de recherche

Environ 1 150 résultats (0,13 secondes)

[PDF] **Document provisoire, ne pas diffuser - Haut Conseil de l...**
www.sante-jeunesse-sports.**gouv.fr**/IMG/pdf/r_mt_300905_vhb5.pdf
de D ANTONA - Autres articles
RAPPORT DU GROUPE DE TRAVAIL du Conseil supérieur d'hygiène publique de France. Risque de contamination horizontale au sein de collectivité d' ...

[PDF] **Guide du bon usage des médias sociaux - Ministère de la ...**
www.defense.**gouv.fr**/guide-medias-sociaux/telecharger.pdf
à **ne pas diffuser**. • Éviter les publications, statuts ou commentaires tels que : « Super ! Plus que 11 jours et 2 heures et vous serez à quai et je pourrai enfin te ...

Getting domains (plan B): using the *cloud*

Wait a second...

- We have a scalable architecture
- We have France == 97M IPs (GeoIP + whois)
- We have liblleeloo to distribute these 97M IPs over our probes

Getting domains (plan B): using the *cloud*

Wait a second...

- We have a scalable architecture
- We have France == 97M IPs (GeoIP + whois)
- We have liblleeloo to distribute these 97M IPs over our probes

⇒ Let's distribute the 97M DNS lookups !!

Results

- Duration: 15h
- Hosts: 5
- Unique domains found: 1342
- Unique IPs: 1295
- Subdomains: 143
- Network size: 7M IPs

Conclusion: targeting *.gouv.fr at cloud age

Finding targets

```
1  def domains2IP( hostnames, patter ):
2      domains = hostnames.grep( pattern ) # 1342 domains
3      targets = []
4      for d in domains:
5          ip = gethostbyaddr( d )
6          targets += net.add( whois.get_range( ip ) )
7      return targets
```

Plan

Plan

A quick word about heartbleed

- Many scans looking for vulnerable servers...
- Most of the focus is on 443 port
- Free advice: people should also look at OpenVPN and some other servers

Looking for a backdoor

I'm gonna owned the Internet

- Backdoor discovered (twice :) by Eloi Vanderbeken on some routers
- Listen on TCP port 32764
- No authentication, simple protocol
- Let's start some recognition...

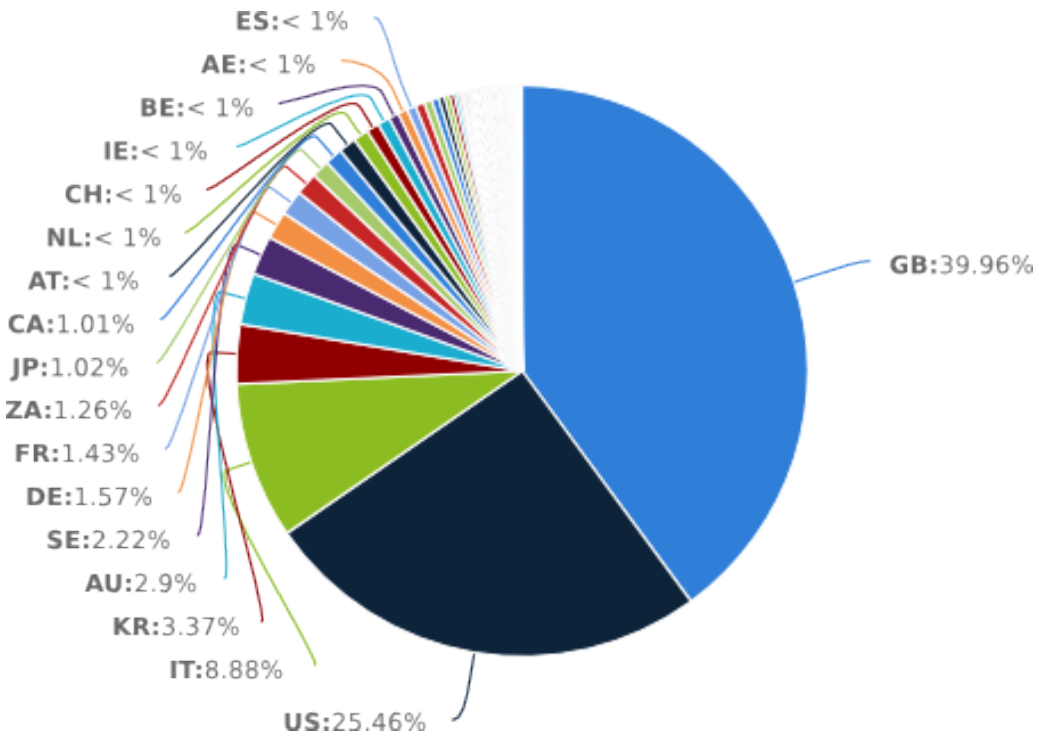
How to own the Internet

My precious

- Launch masscan on 32764: 30k packets/s
- around 50h later, about 1 million IPs discovered with TCP port 32764 open
- Used nodescan to verify these hosts: checking for backdoor signature as an answer of an invalid request
- By scanning about 6k IPs/s, a few minutes later, about 6000 devices were found vulnerable

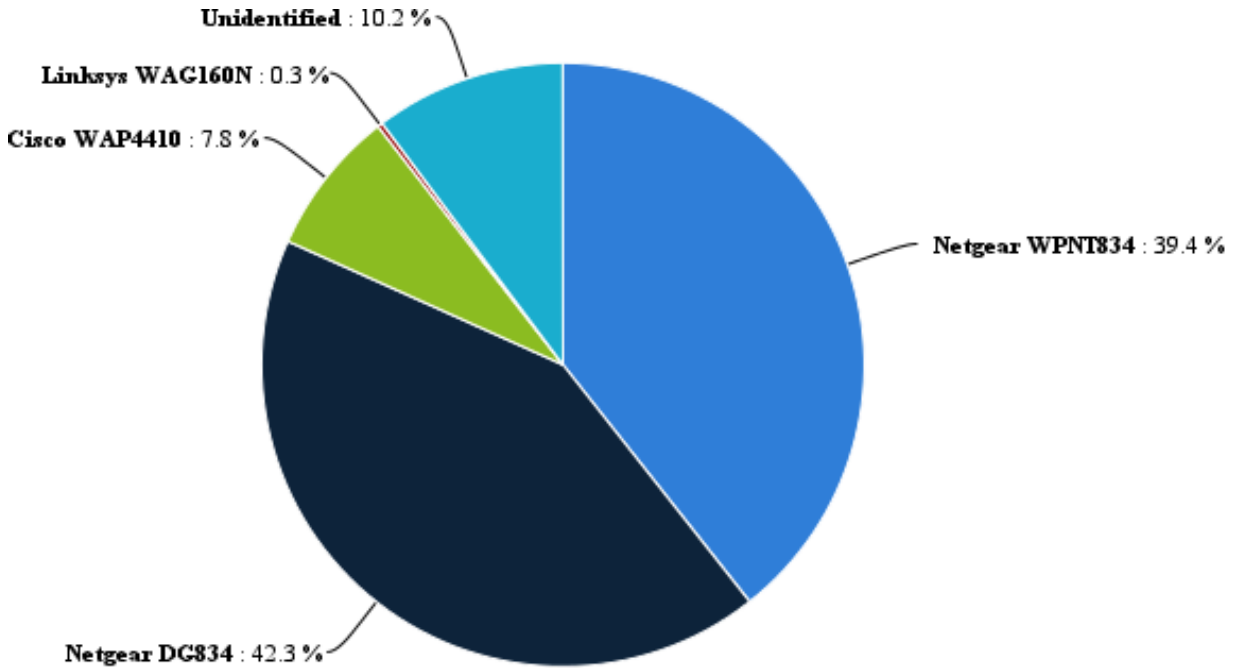
Gathering statistics about the backdoor

- Repartition by country



Gathering statistics about the backdoor

- Repartition by hardware: using the same scanner, a "version" and "sys_desc" field has been grabbed. Manual mapping had to be done (thus the "Unidentified" field).



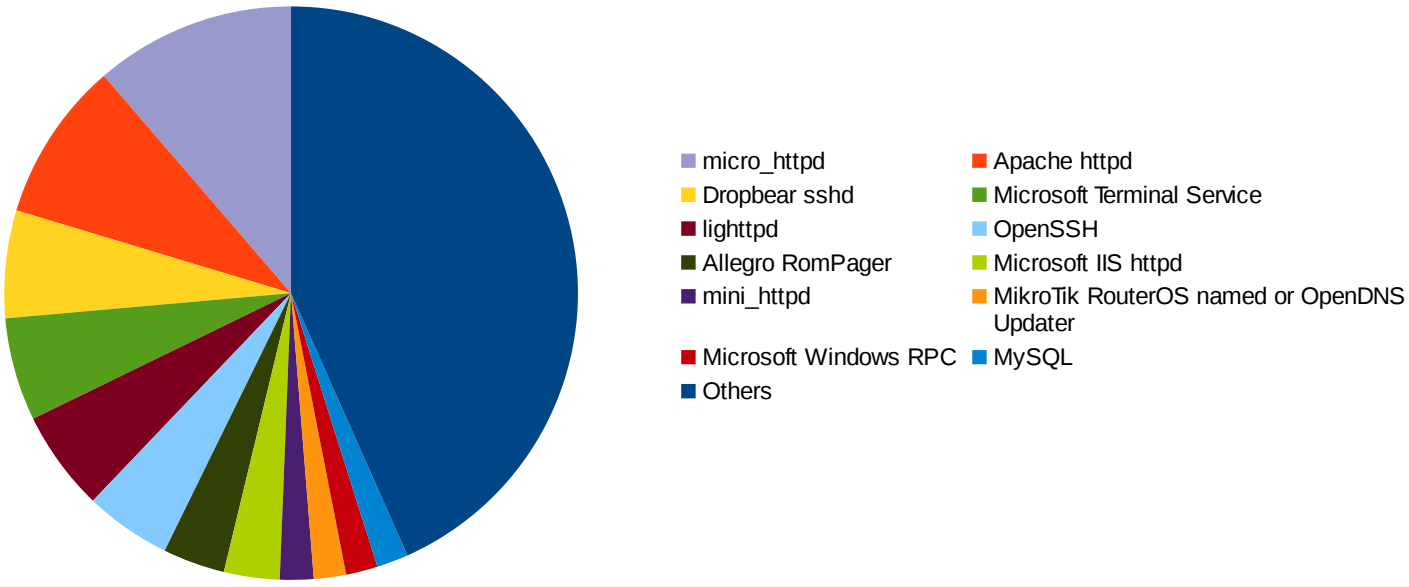
Plan

#define Spain

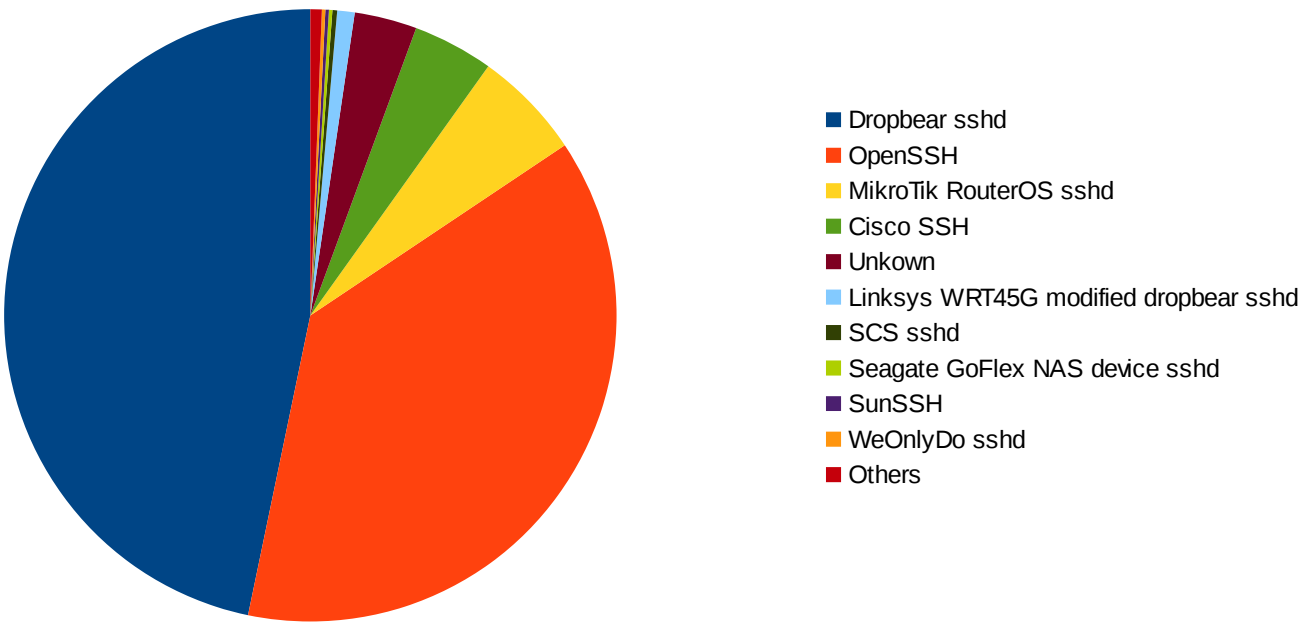
What is Spain?

- Country: 30M IPs
- Number of probes: 100
- Number of ports: 30
- Plugins: banners for Telnet & FTP, SSL certificate, SSH key, HTTP (index of, page title, headers, auth), heartbleed, NFS.ls, FTP.ls, MySQL info, hadoop,...
- Scan duration: 25h

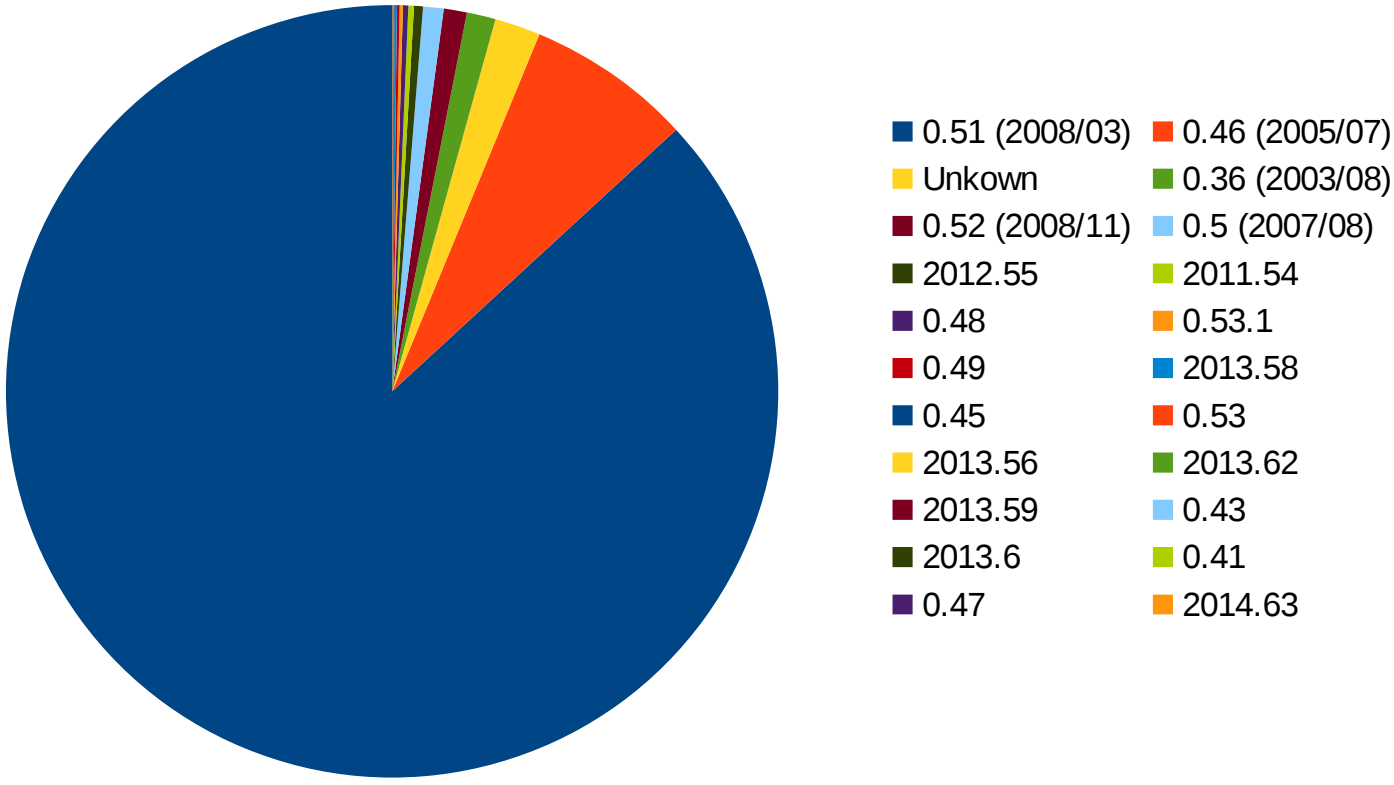
What does Internet.es look like?



Focus on SSH


Dropbear sshd
OpenSSH
MikroTik RouterOS sshd
Cisco SSH
Unkown
Linksys WRT45G modified dropbear sshd
SCS sshd
Seagate GoFlex NAS device sshd
SunSSH
WeOnlyDo sshd
Others

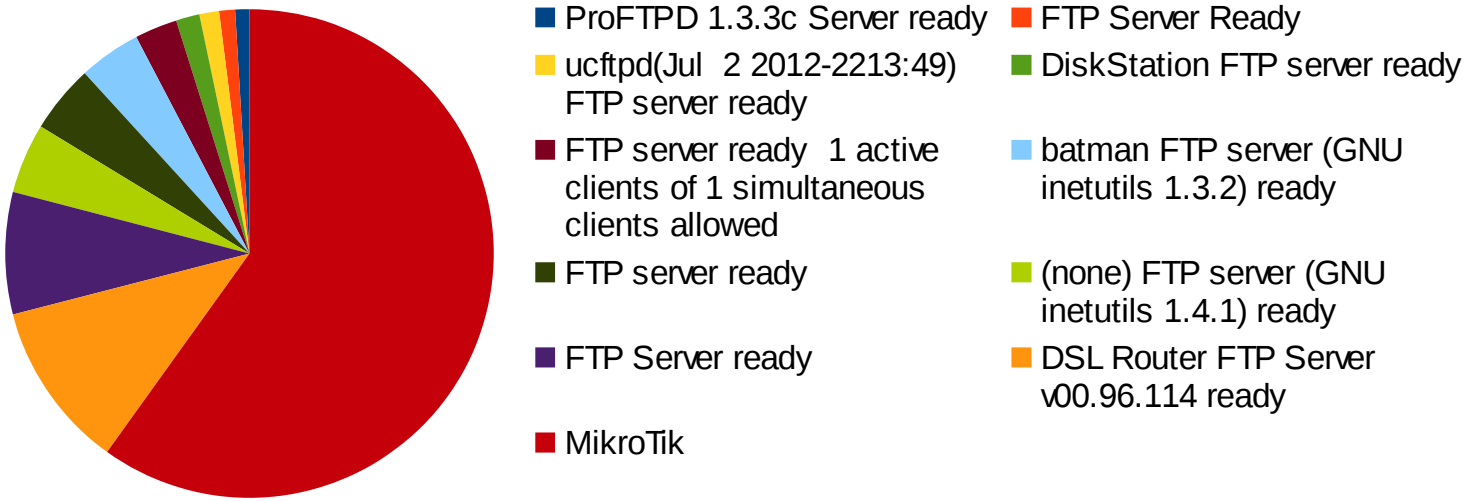
Digging into dropbear




```
37.152.157.114.txt: OLECOMUNICACION-NET 84.232.91.100.txt: THELLINM-NET
37.61.251.139.txt: NUBBITEL 84.236.239.171.txt: ADSLSERVICES-ADSL-NET
62.81.244.73.txt: INFOTEC_TECNOLOGIA_INTEGRAL_Y_TELEC 87.235.106.48.txt: IPCOM-NET
77.27.81.19.txt: FUNDACIONCULTURALESTRADA-NET 89.140.120.114.txt: INFORMATICA_LIMON
```

FTP at a glance

- FTP banners: 31959
- grep -i camera ftps|wc -l → 216
- grep -i "DSL router" ftps|wc -l → 2110



MikroTik FTP?

- Actually all FTP banners containing MikroTik are unique

```
LOURDES GARCIA LANDETE FTP server (MikroTik 5.11) ready: 1
Nodo Formentera 2 V + H FTP server (MikroTik 5.25) ready: 1
AYTO_SCOLA_MUSICA FTP server (MikroTik 5.25) ready: 1
Cliente Danubio27 - Francisco Planells FTP server (MikroTik 5.19) ready: 1
M26002512T FTP server (MikroTik 5.22) ready: 1
SJVJCostaRd1 FTP server (MikroTik 5.22) ready: 1
ramon lopez perez FTP server (MikroTik 5.21) ready: 1
```

Someone is looking at your FTP servers

A long time ago, in a far far away FTP server...

- We noticed a file `w0000000t.{php,txt}` on 115 world-writable FTP servers

```
>> cat w0000000t.txt
w000000000000000t
>> cat w0000000t.php
<?php echo base64_decode("bm9wZW5vcGVub3Bl"); ?>
; nopenopenope
```

- 104 out of the 115 are Microsoft FTPd
- Google(bm9wZW5vcGVub3Bl) $\rightarrow$ 2 servers
- Google(w00...000t) $\rightarrow$ more IPs
- Anyone knows what tool lets this signature?

Long tail of Internet.es (a.k.a. wtf.es)

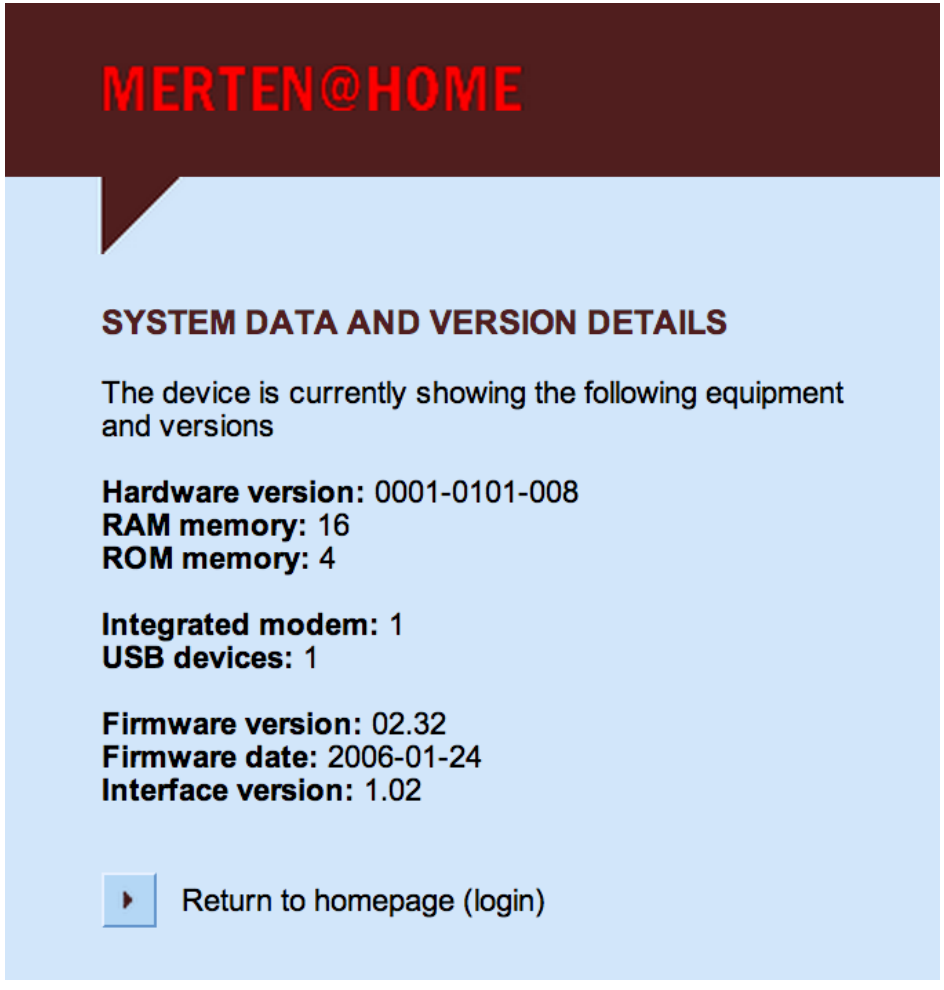
- 3M Filtrete 3M-50 thermostat: thermostat with WiFi control... on the Internet
- http://www.radiothermostat.com/filtrete/products/3M-50/

Long tail of Internet.es (a.k.a. wtf.es)

- merten@home: remote for everything at home

Long tail of Internet.es (a.k.a. wtf.es)

- merten@home: awarded in 2004 and 2006 !!



Long tail of Internet.es (a.k.a. wtf.es)

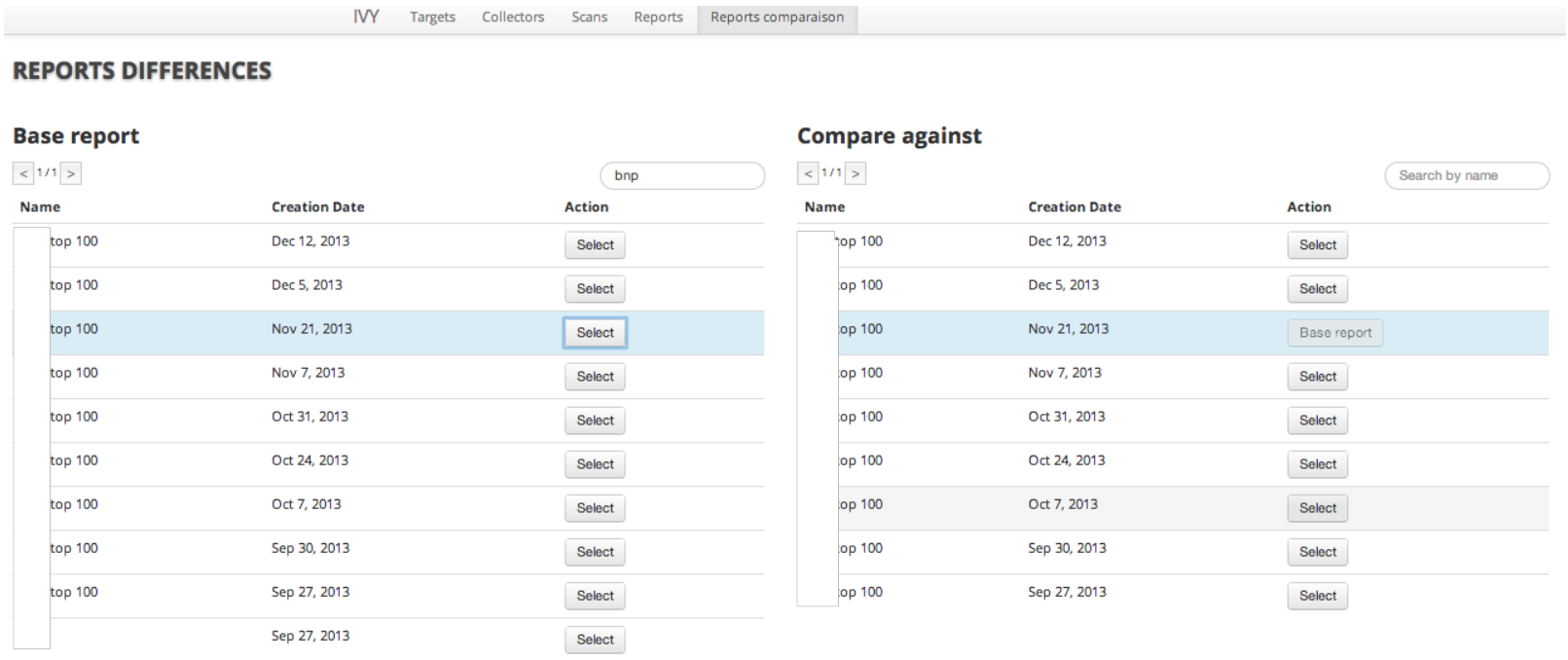
- Moxa NPort 5410: serial to IP converter for PLC, industrial systems, ...

Long tail of Internet.es (a.k.a. wtf.es)

- Cameras of course: Axis

Plan

Monitoring == diffing



Plan

PayTV Internet Sharing

CCcam

- One host (master) shares a card with several clients
- When one client receives an encrypted payload, it is sent to the master
- The master deciphers the payload, sends it back to the client
- Very lucrative business

Usage statistics

- Scan a few ports, the usual ones where CCcam is running
- Connect to the server to get plenty of information

Piracy monitoring

IVY | Targets | Collectors | Scans | Reports | Reports comparaison

DASHBOARD
Summary
DETAILS
View by ip
View by port
View by plugin
PLUGINS
cccam-infos
GRAPHICS
View history

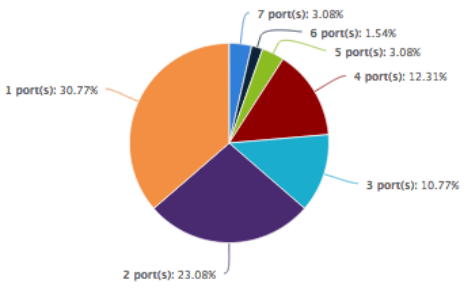
CCCAM FINAL

KEY NUMBERS

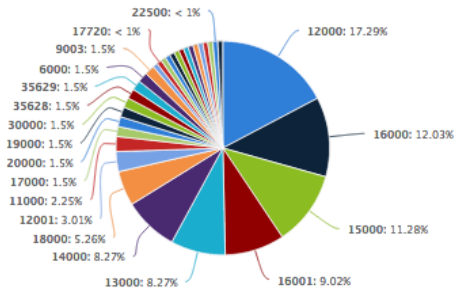
Scanned targets : 65 **Opened targets :** 65 (100%) **Total opened ports :** 133 **Max/Avg opened ports :** 7 / 2

DISTRIBUTION BY PORT NUMBER, COUNTRY AND TLD

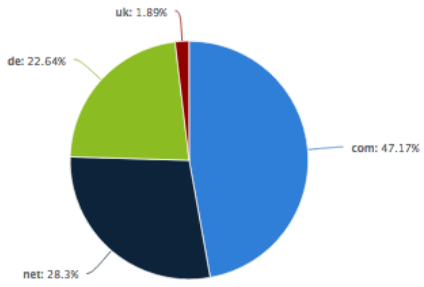
Targets repartition by opened ports
Graph built the 21/01/2014



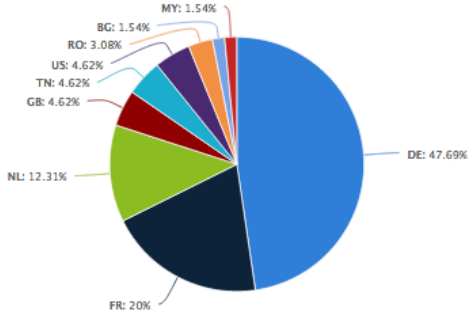
Targets repartition by ports
Graph built the 21/01/2014



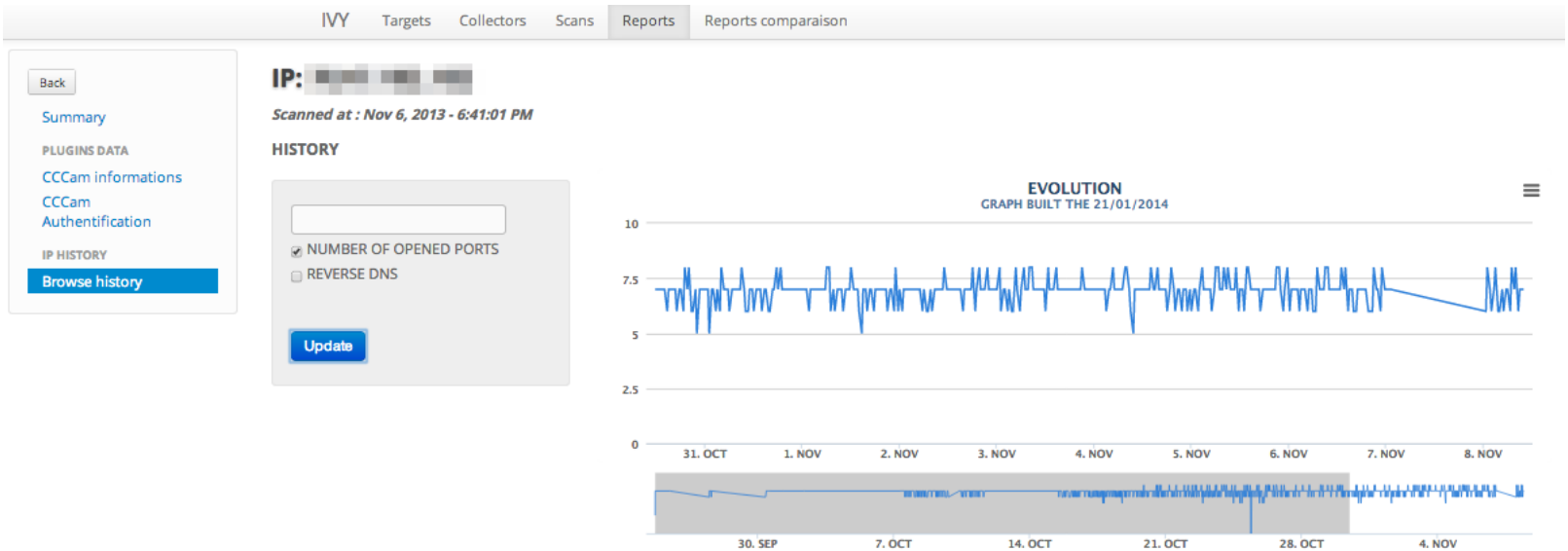
Targets repartition by TLD
Graph built the 21/01/2014



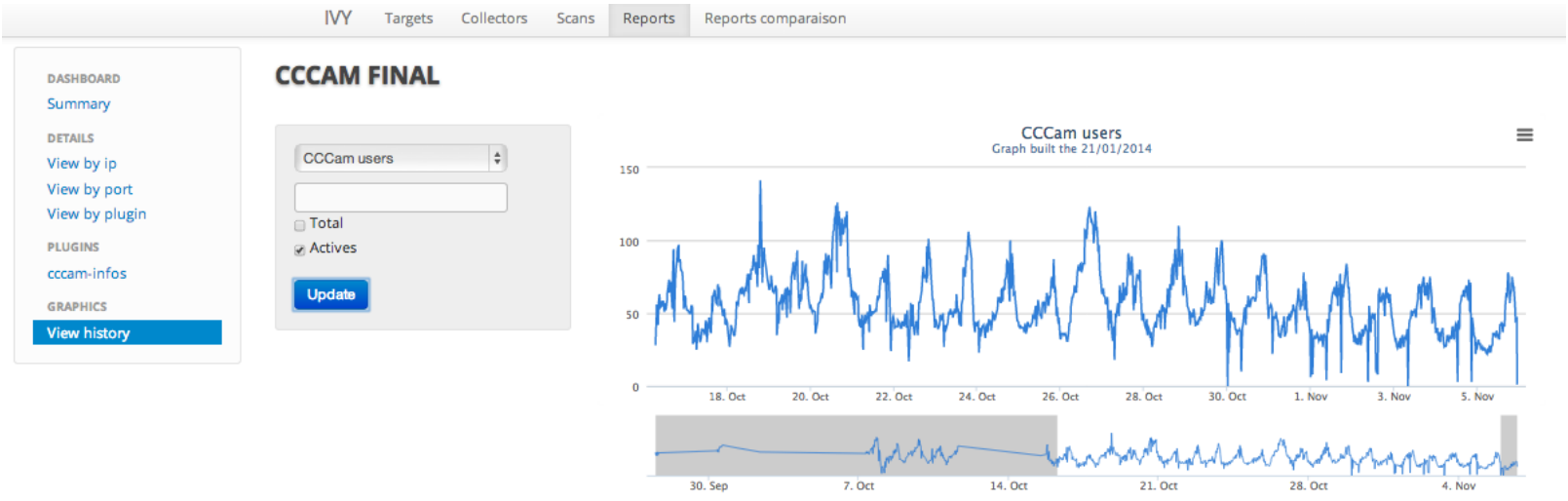
Targets repartition by country
Graph built the 21/01/2014



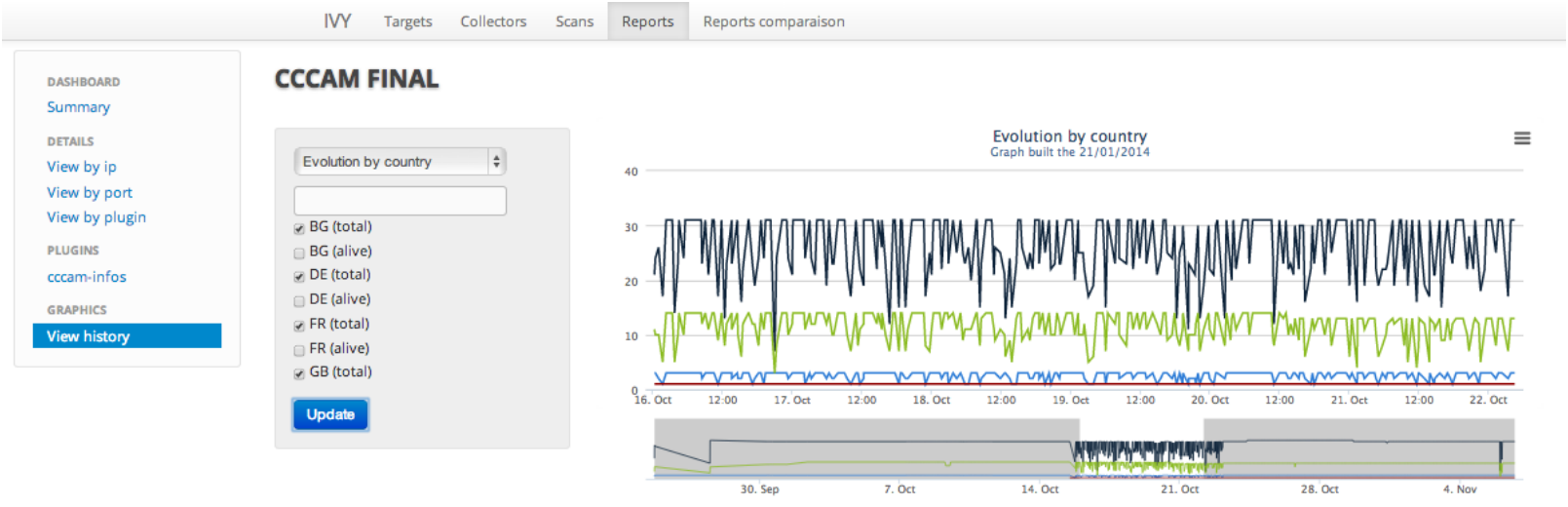
Piracy monitoring



Piracy monitoring



Piracy monitoring



Conclusion

Port scan is not for pussies anymore at the cloud age

- Port scan is not only about the port scanner itself
  - **Scalability**: distribution of the task
  - **Big Data**: unstructured data with a lot of inserts, need for indexation
- **Admin**: sending automatically emails to `abuse@...` is free, but you should have more serious things to deal with that port scans in 2014
- **Legal**: no idea if it is legal or not, but if it is not, it just helps the bad guys, so it is stupid

Conclusion

What massive port scan is good for?

- Security is not about patching anymore
- Try to prevent the attack (ID,PS, exploit mitigation, AV, . . .)
- Assume the attack will succeed anyway :(

⇒ Need to know what / where your assets are

  - To elaborate your defensive strategy
  - To elaborate your recovery plan

Questions?

Challenge accepted:klapspaan

www.quarkslab.com

contact@quarkslab.com | @quarkslab.com